

REC'D 03 JAN 2001

WIPO

PCT

日本国特許庁

PATENT OFFICE  
JAPANESE GOVERNMENT

PCT/JP00/07916

10.11.00

JP00/7916

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日

Date of Application:

1999年11月11日

出願番号

Application Number:

平成11年特許願第321788号

出願人

Applicant(s):

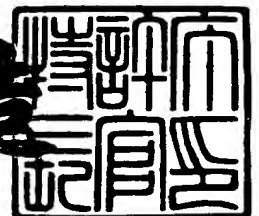
株式会社アイエスデー研究所  
ソフトウェア生産技術研究所株式会社

PRIORITY  
DOCUMENT  
SUBMITTED OR TRANSMITTED IN  
COMPLIANCE WITH RULE 17.1(a) OR (b)

2000年12月15日

特許庁長官  
Commissioner,  
Patent Office

及川耕造



出証番号 出証特2000-3103524

【書類名】 特許願  
【整理番号】 99101  
【提出日】 平成11年11月11日  
【あて先】 特許庁長官殿  
【発明の名称】 プログラム再生方法  
【発明者】

【住所又は居所】 神奈川県鎌倉市十二所 9 6 7 - 6 4

【氏名】 根来 文生

【特許出願人】

【識別番号】 598153401

【氏名又は名称】 株式会社アイエスデー研究所

【特許出願人】

【識別番号】 599086238

【氏名又は名称】 ソフトウェア生産技術研究所株式会社

【代理人】

【識別番号】 100080322

【弁理士】

【氏名又は名称】 牛久 健司

【連絡先】 0 3 - 3 5 9 3 - 2 4 0 1

【選任した代理人】

【識別番号】 100104651

【弁理士】

【氏名又は名称】 井上 正

【連絡先】 0 3 - 3 5 9 3 - 2 4 0 1

【手数料の表示】

【予納台帳番号】 006932

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】	図面 1
【物件名】	要約書 1
【物件名】	委任状 1
【援用の表示】	平成 1 1 年 1 1 月 1 1 日提出の包括委任状
【物件名】	委任状 1
【援用の表示】	平成 1 1 年 1 1 月 1 1 日提出の包括委任状
【物件名】	委任状 1
【援用の表示】	平成 1 1 年 1 1 月 1 1 日提出の包括委任状
【物件名】	委任状 1
【援用の表示】	平成 1 1 年 1 1 月 1 1 日提出の包括委任状
【ブルーフの要否】	要

【書類名】 明細書

【発明の名称】 プログラム再生方法

【特許請求の範囲】

【請求項 1】 Lyeeの開発手法以外の方法で作成された既存のプログラムを、その既存のプログラムが達成する機能を維持して、Lyeeの開発手法に従う構造を持つ新たなプログラムに再生する手法であり、

既存のプログラムの記述に基づいて、すべての単語を、それらの単語が関連する定義体を識別した形で抽出し、

抽出した単語を、その単語が関連する定義体に応じて、一または複数のパレットに割付け、

既存のプログラムの各命令を、その命令に関連する単語が割付けられたパレットの複数のプログラム構成要素のいずれかにしたがつう形に変換する、

プログラム再生方法。

【発明の詳細な説明】

【0001】

【技術分野】

この発明はプログラム再生方法に関し、さらに詳しくはLyeeの開発手法以外の方法で作成された既存のプログラムを、その既存のプログラムが達成する機能を維持して、Lyeeの開発手法にしたがつう構造をもつプログラムを再構築する方法に関する。

【0002】

【背景技術】

Lyee (governmental methodology for software providence) のプログラム開発手法は従来のソフトウェア作成方法を根底から変革するものとして登場し、開発期間の飛躍的短縮、開発工数の大幅な減少、保守の容易さ、ドキュメント量の減少等の画期的な成果をあげるものとして注目されている。

【0003】

従来のプログラムはシステム・エンジニアやプログラマーの能力に依存し、個

性が出やすい。そのために、既存のプログラムを解析するのはたとえサポート・ドキュメントがあったとしても容易ではないどころか、不可能に近い場合もある。そのために、プログラムの動作環境を変更（コンピュータ・マシンやOSの変更）しようとする場合には、結局のところ既存のプログラムを捨てて、新たにプログラムを作成するより方法はないというのが現実であった。

#### 【0004】

##### 【発明の開示】

この発明は、既存のプログラムを解析し、その解析結果に基づいて既存のプログラムと同等の機能を達成するプログラムに生まれ変わらせる新しい方法を提供するものである。再生されるプログラムはLyeeの方法論にしたがうものであり、再生の処理をできるだけ自動化できるようにするものである。

#### 【0005】

この発明による方法は、Lyeeの開発手法以外の方法で作成された既存のプログラムを、その既存のプログラムが達成する機能を維持して、Lyeeの開発手法にしたがう構造をもつ新たなプログラムに再生する方法であり、既存のプログラムの記述に基づいて、すべての単語を、それらの単語が関連する定義体を識別した形で抽出し、抽出した単語を、その単語が関連する定義体に応じて、一または複数のパレットに割付け、既存のプログラムの各命令を、その命令に関連する単語が割付けられたパレットの複数のプログラム構成要素のいずれかにしたがう形に変換するものである。

#### 【0006】

後に詳述するように、Lyeeの方法論にしたがうプログラム（基底論理：プログラム構成要素）は、一つの系を形成するW04, W02, W03パレットごとに、単語を基本として設けられるものである。この発明によると、既存のプログラムから、すべての単語を抽出しているから、Lyeeのプログラム構造を満たすものであり、かつ単語の抽出作業をほぼ自動化できる。また、抽出した単語を複数のパレットに割付けている。この割付けも自動化が容易である。既存のプログラムの各命令がLyeeの方法論にしたがうプログラムに対応づけられて変換されることにより、再生されたプログラムは既存のプログラムと同じ機能を達成する。

【0007】

このようにして、この発明によると、既存のプログラムをLyeeのプログラムの形に比較的容易にかつほぼ自動的に再構築できる。したがって、既存のコンピュータに搭載されている既存のプログラムを、新たなコンピュータに、同じ機能を達成する新たなプログラムとして比較的容易に移植することが可能となり、プログラム環境の変化に対応できる。しかも、Lyeeの方法論にしたがうものであるから、保守等も容易である。

【0008】

この発明は、上述したプログラム再生方法をコンピュータに実行させるための再生プログラムを記録した媒体および再生方法を実現する装置（再生プログラムがインストールされたコンピュータ）も提供する。

【0009】

【実施例の説明】

(1) Lyeeの方法論にしたがうソフトウェア・モデル

コンピュータは人間が意味を捉えることを助ける道具である。人間が意味を捉え、判断し、行動を起こすためには、まず情報が必要である。コンピュータは情報（情報が無いことも含む）を人間に対してアウトプットする。これが出発点となる。コンピュータのこの出力部分（出力に関するソフトウェア部分）をW04パレットと呼ぶ（パレット：意味を運ぶものという意味）。

【0010】

コンピュータの出力を見て人間はその意味を認識し、判断し、行動する。人間の行動はコンピュータに対しては入力という形で具体化される。コンピュータの入力に関するソフトウェア部分をW02パレットと言う。

【0011】

コンピュータは入力に応答してあらかじめ定められた処理を実行する。この処理がソフトウェアの核心的な部分となる。入力に応答してコンピュータが行う処理に関するソフトウェア部分をW03パレットとする。

【0012】

コンピュータは処理の結果を再び出力する。すなわち、W03パレットの次に

W04パレットが順序づけられる。このようにして、Lyeeの方法論にしたがうと、ソフトウェア・モデルは、基本的には、W04パレットから出発し、W02パレット、W03パレットを経て再びW04パレットに至るという循環の構造を持つ。この循環モデルが図1に示されている。

#### 【0013】

認識の対象である情報は、人間が視覚的かつ直接的に認識できる媒体に限られる。これらの媒体は、たとえば、画面、帳票などである。入力に関するW02パレットおよび出力に関するW04パレットがこれらの媒体を取り扱う。また、処理に関するW03パレットが取り扱う媒体には画面、帳票などに加えてファイル、データベース、電文等が含まれる。これらの画面、帳票、ファイル、データベース、電文等を定義体と名付ける。

#### 【0014】

複数種類の画面が存在するときに、これらを識別するための符号が必要である。W02パレットおよびW04パレットにおいて、定義体を識別する符号を定義体識別子 $k$ と置く。またW03パレットにおいて、定義体を識別するための符号を定義体識別子 $d$ と置く。

#### 【0015】

画面、ファイル、データベース等の定義体には、さまざまなデータ項目（データフィールド）が現われる。データ項目のようなソフトウェアが扱うデータを管理するための最も基本的な単位を単語という。単語はこれ以上分けられない最小の単位である。たとえば、年月日という項目は、年という単語、月という単語および日という単語に分けられる。単語には、文字、数字、コマンドが含まれる。単語を識別するための識別符号を、単語識別子 $i$ という。

#### 【0016】

Lyeeの方法論によると、単語を単位とした最もシンプルな構造のプログラム（手順）が用意される。この基本的なプログラムを基底論理と呼ぶ。基底論理はパレットごとに設けられる。基底論理には、論理要素（関数）、位相要素（関数）および作用要素（関数）がある。

#### 【0017】

W 0 3 パレットの論理要素を  $L_3(d, i)$  , 位相要素を  $Y_3(k, i)$  , 作用要素を  $P_3(j)$  と置く。ここで  $j$  は作用子である。

【0 0 1 8】

W 0 4 パレットの論理要素を  $L_4(k, i)$  , 位相要素を  $Y_4(k, i)$  , 作用要素を  $P_4(j)$  と置く。

【0 0 1 9】

W 0 2 パレットには論理要素のみがあり、これを  $L_2(k, i)$  と置く。

【0 0 2 0】

W 0 3 パレットの論理要素 ( $L_3(d, i)$  ) のプログラムをフローチャートで図 2 に示す。「空か?」というのは、定義体  $d$  の単語  $i$  についての W 0 3 パレットのデータフィールドにデータが存在しないかをチェックするもので、データが存在すればもはや処理を行なう必要がないので、終了する。データが存在しなければ、「自己生成」処理、すなわちこの単語  $i$  に特有の処理（たとえば、MOV 文にしたがう処理、演算処理など）を実行する。この処理が実行できれば「成立」となる。実行できなければ、処理に必要なデータが無いのであるから、「不成立」であり、「再起カウンタ + 1」（再起カウンタのインクリメント）をする。再起カウンタの値が所定値に達していなければ（「再起?」で YES）, 「再起フラグをセット」して、再度この論理要素に戻るようにする。再起カウンタの値が所定値以上になれば、「不成立フラグをセット」する。この場合には、一般にエラーメッセージが論理要素  $L_4(k, i)$  により編集される。

【0 0 2 1】

図 3 は W 0 2 パレットの論理要素  $L_2(k, i)$  のプログラムのフローチャートである。W 0 2 パレットは単語の属性をチェックする役割を持つものである。「属性検査」は、数字か、文字か、コマンドかのチェックを行うものである。

【0 0 2 2】

図 4 は W 0 4 パレットの論理要素  $L_4(k, i)$  のプログラムのフローチャートである。W 0 4 パレットは W 0 3 パレットの結果を「表示」のために要求された形に「編集」して出力する役割を担う。

【0 0 2 3】



これらのW 0 4 論理要素, W 0 2 論理要素およびW 0 3 論理要素のプログラムの一例を, 図5に示す。□(空欄)の部分に定義体識別子k(またはd)および単語識別子iが代入される。

#### 【0 0 2 4】

位相要素 $Y_3(k, i)$ ,  $Y_4(k, i)$ はW 0 2パレットの結果をW 0 3パレットに, W 0 3パレットの結果をW 0 4パレットに, それぞれ結合させる役割を持つもので, データの複写(転送)を行う。

#### 【0 0 2 5】

Lyeeの方法論によると, 上述したように, 一つの系が3種類のW 0 4, W 0 2, W 0 3パレットによって構成されており, この一つの系が一つのデータを単位とする処理である。これらの系のつながりを表現したのが処理経路図である。

#### 【0 0 2 6】

コンピュータの出力(W 0 4パレット)は一般に表示画面等を通して行われ, この画面を見て人間が行動(入力)することにより, W 0 2パレットによる処理へとつながる。このように, 人間の行動が処理の契機となるものをオンライン処理T 0と呼ぶ。これに対して, ファイルという情報を記憶する媒体を基盤とするデータの性格で動く処理をバッチ処理T 1という。

#### 【0 0 2 7】

オンライン処理T 0の処理経路図の一例を図6に示す。この処理経路図には, W 0 4パレットa 1, W 0 2パレットa 2およびW 0 3パレットa 3から構成される系と, W 0 4パレットa 1, W 0 2パレットa 2およびW 0 3パレットa 4から構成される系と, W 0 4パレットb 1, W 0 2パレットb 2およびW 0 3パレットb 3から構成される系の3つが含まれている(W 0 4パレットa 1とW 0 2パレットa 2は, W 0 3パレットa 3を含む系およびW 0 3パレットa 4を含む系の2つに共通である)。なお, W 0 3パレットa 3, a 4およびb 3は同じものである。

#### 【0 0 2 8】

W 0 4パレットa 1とW 0 2パレットa 2の間には, 表示装置の表示画面を表すブロックg 1(表示画面g 1という)がある。W 0 4パレットa 1によって表

示画面（画像）g 1 上に文字，数字等が表示される。この表示をみて人間が入力装置を用いて，文字，数字等を入力する。入力されたデータはW 0 2 パレット a 2 によって取り扱われる。同様に，W 0 4 パレット b 1 とW 0 2 パレット b 2 の間には，表示画面 g 2 を表すブロックがある。

【 0 0 2 9 】

表示画面 g 1 には，コマンド・ボタン（「実行」ボタン，「登録」ボタン等）が表示されている。これらのコマンド・ボタンの選択に応じて分岐（分岐点11，13）が生じる。すなわち，分岐点11においては，「実行」ボタンが押されると，W 0 2 パレット a 2 からW 0 3 パレット a 3 に進み，「登録」ボタンが押されると，W 0 2 パレット a 2 からW 0 3 パレット a 4 に進む。W 0 2 パレットからW 0 4 パレットに進むつながりを継続連鎖と呼ぶ。

【 0 0 3 0 】

W 0 3 パレット a 3 の出力側では，W 0 3 パレット a 3 におけるプログラムの記述に応じて，W 0 4 パレット a 1 に戻る経路またはW 0 4 パレット b 1 へ進む経路のいずれかに分岐する（分岐点12）。W 0 3 パレット b 3 の出力側でも同様に，W 0 3 パレット b 3 におけるプログラムの記述に応じて，W 0 4 パレット b 1 への経路またはW 0 4 パレット a 1 への経路のいずれかに分岐する（分岐点14）。一つの系において，W 0 3 パレットからW 0 4 パレットに進むつながりを再起連鎖と呼ぶ。一の系のW 0 3 パレットから他の系のW 0 4 パレットに進むつながり（W 0 3 パレット a 3 からW 0 4 パレット b 1 ）を重複連鎖と呼ぶ。2 つ以上の系を経てもとの系のW 0 4 パレットに進むつながり（W 0 3 パレット b 3 からW 0 4 パレット a 1 ）を多重連鎖と呼ぶ。

【 0 0 3 1 】

W 0 3 パレット a 4 では，データ・ベースA，B，CおよびDを用いたデータ入出力処理（バッチ処理T 1）が行われる。このようなバッチ処理については，図 7 に示すように，一般にさらに詳細な処理経路図を描くことができる。

【 0 0 3 2 】

図 7 に示す処理経路図T 1 には，W 0 4 パレット c 1，W 0 2 パレット c 2 およびW 0 3 パレット c 3 から構成される系と，W 0 4 パレット d 1，W 0 2 パレ

ット d 2 および W 0 3 パレット d 3 から構成される系と、W 0 4 パレット e 1、W 0 2 パレット e 2 および W 0 3 パレット e 3 から構成される系の 3 つが含まれている。

【0033】

W 0 2 パレット c 2 によってデータ・ベース（ファイル）A から読出されたデータが W 0 3 パレット c 3 によってデータ処理され、その処理結果を表すデータが W 0 4 パレット c 1 によってデータ・ベース D に格納される。同様に、W 0 2 パレット d 2 によってデータ・ベース B から読出されたデータが W 0 3 パレット d 3 によってデータ処理され、その処理結果を表すデータが W 0 4 パレット d 1 によってデータ・ベース D に格納される。W 0 2 パレット e 2 によってデータ・ベース C から読出されたデータが W 0 3 パレット e 3 によってデータ処理され、その処理結果を表すデータが W 0 4 パレット e 1 によってデータ・ベース D に格納される。なお、W 0 3 パレット c 3、d 3 および e 3 は同じものであり、W 0 4 パレット c 1、d 1、e 1 は同じものである。

【0034】

W 0 2 パレット c 2 の出力側の分岐点 21 では、データ・ベース（ファイル）A のデータの読出しが終了したかどうか（EOF（End Of File）であるかどうか）によって経路が選択される。EOF であると、W 0 4 パレット e 1 に進む。それ以外では W 0 3 パレット c 3 に進む。

【0035】

W 0 3 パレット c 3 の出力側の分岐点 22 では W 0 3 パレット c 3 におけるプログラムの記述にもとづいて、W 0 3 パレット d 3 の出力側の分岐点 23 では W 0 3 パレット d 3 におけるプログラムの記述にもとづいて、W 0 3 パレット e 3 の出力側の分岐点 24 では W 0 3 パレット e 3 におけるプログラムの記述にもとづいて、2 つに分岐する経路のうちいずれの経路に進むかがそれぞれ選択される。

【0036】

このような処理経路図を作成することは Lyee の方法論では必ずしも必要ではないが、分かりやすくするために説明した。

【0037】

作用要素には経路作用要素，コマンド作用要素，業務作用要素および構造作用要素がある。経路作用要素は上述した分岐等のパレット間の関係を規定するものである。コマンド作用要素はI/Oコマンド（OPEN，READ，WRITE，CLOSE）を取扱うものである。業務作用要素は，データ値の業務的な妥当性チェックを行うものである。構造作用要素は使用済のメモリ・エリアを初期化するものである。これらの作用要素のプログラム構造は図2～図4に示す論理要素のプログラム構造とほぼ同じであるが，詳細については具体例で後述することになる。

## 【0038】

特定のタスクを実行するようにLyeeの方法論にしたがって組立てられた一本のソフトウェアは，それぞれがW04，W02，W03パレットからなる複数の系の集合であり，これらが上述した処理経路図T0またはT1によって表わされるように組合わされている。このような複数の系の集合であるソフトウェアを図8または図9に示すシナリオ関数と呼ぶ関数によって統一的に記述することができる。図8はオンライン処理のシナリオ関数T0を，図9はバッチ処理のシナリオ関数T1をそれぞれ表わしている。{ }は集合を表わす。

## 【0039】

図8において，W02パレット，W03およびW04にそれぞれパレット関数 $\Phi_p(k, 2)$ ， $\Phi_p(3)$  および $\Phi_p(k, 4)$  が作用する。これらのパレット関数は，各パレット内の定義体および単語ごとのプログラム（各要素）を起動条件にしたがって起動するものである。パレット連鎖関数 $\Phi_0$ は各W04パレット，W02，W03を順番に動かして統制をしたり，外部（たとえば表示装置）との通信機能やシステム全体の動作開始，閉塞を制御するものである。

## 【0040】

図9において， $L_2(r, i)$ ， $L_3(r, i)$ ， $L_3(f, i)$ ， $L_4(f, i)$  は論理要素を， $P_2(r)$ ， $P_3(j)$ ， $P_4(f)$  は作用要素を示している。ここで $r$ ， $f$ はそれぞれファイル（定義体）を示し， $r$ は入力側， $f$ は出力側であることを表わす。 $\Phi_p(r, 2)$ ， $\Phi_p(3)$ ， $\Phi_p(f, 4)$  はパレット関数を， $\Phi_1$  はパレット連鎖関数をそれぞれ示す。

## 【0041】

## (2) 既存のプログラムのLyeeのプログラムへの再構築

図10はLyeeの開発手法以外の手法で作成された既存のプログラム（以下、従来プログラムという）を、その従来プログラムが達成する機能を維持して、Lyeeの開発手法にしたがう新たな構造を持つプログラムに再生する処理の全体的な流れを示す流れ図である。プログラム再生処理（プログラム）は再生準備情報作成処理（プログラム）31，解析テーブル作成（プログラム）32，テンプレート適用（プログラム）33，同期化データ構造作成（プログラム）34およびメタTシナリオ関数作成（プログラム）35の順に実行される。これらの処理の一部をコンピュータによる自動化処理ではなく、手作業で行なうこともできる。

## 【0042】

図11は再生準備情報作成処理31の詳細を示すものである。この処理の多くはコンピュータにより自動的に行うことができる。

## 【0043】

はじめに従来プログラムを多階層構造から単階層構造に変換する処理が行われる（ステップ41）。多階層プログラムから単階層プログラムへの変換処理には、次の2つの処理が含まれる。

## 【0044】

その一は、従来プログラムにサブルーチンがあり、メインルーチン中にサブルーチンを呼び出す命令（サブルーチンCALL）が含まれている場合に、メインルーチン中のサブルーチンCALLに代えてそのサブルーチンの命令列をそのまま展開（挿入）する処理である。

## 【0045】

その二は、集団項目（たとえば、「年月日」）を利用して記述されている命令文を、その集団項目を構成する最小の単位の項目（「年」，「月」，「日」の「単語」にまで分解する）を用いた複数の命令文に分解する処理である。

## 【0046】

たとえば、従来プログラムには図12(a) に示すような集団項目とその集団項目を構成する最小の単位の項目（単語）との関係を示す記述がある。

## 【 0 0 4 7 】

このような記述を参照して、従来プログラムの一つの命令文を複数の命令文に分解する。

## 【 0 0 4 8 】

次に、単語の抽出が行われる（ステップ42）。従来プログラムには、Lyeeの開発手法の観点からみると、定義体に属する単語と、媒体アクセスに関連して必然的に必要とされる単語と、従来のプログラマが自分の考えに基づいて自ら定義した単語の3種類に分類される単語がある。

## 【 0 0 4 9 】

従来プログラムとしてコボル（COBOL）言語で記述されたプログラムを例にとる。コボルによるプログラムの構成要素が図13に示されている。

## 【 0 0 5 0 】

コボル・プログラムの具体例が図14に示されている。ここでは、図13に示す構成要素のうち、プログラム識別子の定義、プログラム稼働環境定義、データ・フィールドの定義のうちの連絡節の定義が省略されている。また、図14に示す従来プログラム（コボル言語）の例は、上記の多階層から単階層への変換の処理（ステップ41）により単階層化されたものである。この図において、左欄がデータディビジョン（DATA DIVISION）（データ・フィールドの定義）であり、右欄がプロシジャー・ディビジョン（PROCEDURE DIVISION）（処理手続きの定義）である。プロシジャー・ディビジョンには、説明の便宜上、命令文番号（01～11）を付す。

## 【 0 0 5 1 】

データ・ディビジョン内のファイル・セクション（FILE SECTION）はファイルの定義を与えるものであるから、そこにはファイル名と、ファイルに属するデータ項目が記述されている。

## 【 0 0 5 2 】

ファイル名はファイル（媒体）アクセスに関連して必然的に必要とされる単語を抽出するのに役立つ。

## 【 0 0 5 3 】

ファイルに属するデータ項目は、まさに定義体に属する単語を表わしている。図14に示す例でみると、定義体（ファイルIN-F）に属する単語として「IN-R」が、定義体（ファイルOT-F）に属する単語として「OT-R」が抽出される。単語の抽出処理とともに定義体の識別も行なわれる。

#### 【0054】

ファイル・セクション内のファイル名が認識される。図14の例では「IN-F」と「OT-F」である。これらのファイル名（媒体名）は媒体アクセス命令検出用として用いられる。すなわち、プロシジャー・ディビジョンに記述された命令の中から、媒体アクセスコマンド命令検出用ファイル名（媒体名）を使用している命令だけを抽出する（コマンド一覧表の作成）。このようにして抽出された命令から、媒体アクセスに関連して必然的に必要とされるデータ項目（単語）が抽出される。

#### 【0055】

たとえば、プロシジャー・ディビジョンの命令文番号03から、図15に示すように、「IN-EOF-F」が媒体アクセスに関連して必然的に必要とされるデータ項目（単語）として抽出される。

#### 【0056】

ワーキング・ストレージ・セクション（WORKING-STORAGE SECTION）に記述された単語のうち、定義体に属する単語および定義体へのアクセスに関連して必然的に必要とされる単語として抽出されていない単語（「WK」および「WK2」）が、プログラマが任意に定義した単語として抽出される。

#### 【0057】

処理経路図を作成する（ステップ43）（図6および図7参照）（処理経路図の作図は、上述のように必ずしも必要なものではない）。また、この処理は一般的には手作業で行なわれる。

#### 【0058】

先に説明したように、オンライン処理T0については表示画面（スクリーン）を中心として処理経路図を描くことができ、バッチ処理T1についてはファイルを中心に処理経路図を描くことができる。これらのスクリーン、ファイルは媒体

である。

#### 【0059】

そこで、媒体アクセスコマンド命令検出用ファイル名（媒体名）を使用している命令だけを抽出して作成したコマンド一覧表を処理経路図の作成に利用することができる。

#### 【0060】

図14に示す従来プログラムの例はファイル処理に関するものであるから、バッチ処理T1の例である。バッチ処理においては、出力系のアクセス命令（WRITE等）はW04パレットに関係し、入力系のアクセス命令（READ等）はW02パレットに関係する。したがって、W04パレットをWRITEされるファイルごとに設定し、W02パレットをREADされるファイルごとに設定し、他の命令に関してはW03パレットを無条件に設定すれば処理経路図が描かれる。

#### 【0061】

また、この作業により、各媒体（定義体）がどのパレットに関係するかが明らかになるから、媒体にそれが関係するパレットの符号を付し、パレット識別子を定める。たとえば、命令文番号03において、「READ IN-F」においては、ファイルIN-FにIN-F-W02というパレット識別子を付す。

#### 【0062】

抽出された単語のパレットへの割付が行われる（ステップ44）。

#### 【0063】

先に「(1) Lyeeの方法論にしたがうソフトウェア・モデル」で説明したように、Lyeeの方法論によると、単語を単位とした最もシンプルな構造のプログラム（手順）が用意される。この基本的なプログラムを基底論理と呼ぶ。基底論理はパレットごとに設けられる。基底論理には、論理要素（関数）、位相要素（関数）および作用要素（関数）がある。

#### 【0064】

この章の「(2) 既存のプログラムのLyeeのプログラムへの再構築」の目的は、Lyeeの方法論による単語を単位とした最もシンプルな基本的なプログラムである基底論理を、既存のプログラムの達成する機能を維持して構築することにある。



基底論理はパレットごとに設けられるから、すべての単語をパレットに割付けることが必要である。一つの単語は一般的には2つ以上のパレットに割付けられる。

#### 【0065】

まず、ラショナルリティ (Rationality) について説明しておく。図16は、ラショナルリティの一般構造を示すもので、全体的にみると処理経路図 (シナリオ関数) T0 (一つの系のみを示す) がある。処理経路図T0はT0W02パレット、T0W03パレットおよびT0W04パレットからなる構造をもつ。

#### 【0066】

T0W03パレット内に処理経路図 (シナリオ関数) T1がある。T0W02パレットからT0W03内の「T0入力」を経て処理経路図T1に至り、処理経路図T1からT0W03内の「T0中間」 (T0出力) を経てT0W04パレットに至る。

#### 【0067】

処理経路図T1はT1W02パレット、T1W03パレットからT1W04パレットに結ばれている。T1W03パレット内に「T1入力」と「T1中間」が設けられる。

#### 【0068】

さらに、T0W03パレット内には、処理経路図t1を設ける。処理経路図 (付加的シナリオ関数) t1は、先に説明したように、従来プログラムではプログラマが任意に定義する単語が存在し、この単語にかかわる処理をLyeeの手法にしたがうソフトウェア構造に再構築するために必要となるものである。T0W03パレット内においては、「T0入力」から処理経路図t1に至り、処理経路図t1から「T0中間」と「T1中間」とに至る。

#### 【0069】

処理経路図t1内においても、t1W02パレット、t1W03パレットおよびt1W04パレットがある。t1W03パレット内にも「t1入力」と「t1中間」がある。

#### 【0070】

前述したように、単語は、定義体に属する単語、媒体アクセスに関連して必然的に必要とされる単語、および従来のプログラマが自分の考えに基づいて自ら定義した単語の3種類がある。

#### 【0071】

定義体に属する単語のうち画面（スクリーン）に属する単語が処理経路図T0に関係し、ファイルに属する単語が処理経路図T1に関係する。媒体アクセスに関連して必然的に必要とされる単語を割付ける（取扱う）ために、コントロール・ボックス（C/B）が設けられる。プログラマが定義した単語は処理経路図t1に割付けられる。

#### 【0072】

処理経路図T0およびT1に関係する単語、すなわち定義体に属する単語のパレットへの割付について説明する。

#### 【0073】

W02パレットへの単語の割付は次の通りである。先に説明したように、定義体（ファイルなど）にはパレット識別子が付される。符号W02が付されたパレット識別子をREAD定義体識別子とすると、READ定義体識別子をもつ定義体に属する単語を基本としてW02パレットにおける基底論理をつくる。基底論理ごとのデータフィールドもW02パレットのデータ・ディビジョンに定義する。さらに、バッファエリア（WFL）も設ける。

#### 【0074】

W04パレットへの単語の割付は、同じようにWRITE定義体識別子（符号W04が付されたもの）をキーにして、先に抽出された単語をW04パレットにおいて実現する基底論理の基本単語とする。その基底論理ごとのデータフィールドもW04パレットのデータ・ディビジョンに定義する。さらに、バッファエリア（WFL）も設ける。

#### 【0075】

W03パレットへの単語の割付は、W02パレットに割り付けた単語をそのまま「W03パレットの入力」に割付、W04パレットに割り付けた単語をそのまま「W03パレットの中間」に割付る。

## 【 0 0 7 6 】

プログラマが定義した単語は処理経路図 t 1 内の W 0 2 , W 0 4 , W 0 3 入力および W 0 3 中間に割付ける。

## 【 0 0 7 7 】

ファイル・アクセスのために必然的に必要な単語はコントロール・ボックスに割付ける。

## 【 0 0 7 8 】

図17は、図14に示す従来プログラムの記述から抽出された単語が各パレットに割付けられた状態を表わすものである。

## 【 0 0 7 9 】

抽出された単語は、定義体の単語 (IN-R, OT-R) , プログラマが定義した単語 (WK, WK2 ) , 定義体へのアクセスに必要とされる単語 (IN-EOF-F) に分類される。

## 【 0 0 8 0 】

単語 IN-R は定義体 (ファイル) IN-F の単語であり、ファイル IN-F は入力側のファイルである。単語 IN-R は、処理経路図 T 1 の W 2 パレットに割り付けられる。

## 【 0 0 8 1 】

単語 OT-R は定義体 (ファイル) OT-F の単語であり、ファイル OT-F は出力側のファイルである。単語 OT-R は処理経路図 T 1 の W 0 4 パレットに割り付けられる。単語 IN-F とファイル IN-F との間、単語 OT-R とファイル OT-F との間バッファ・エリア (W F L) が設けられる。

## 【 0 0 8 2 】

処理経路図 T 1 の W 0 2 パレットに割り付けられた単語と同じ単語が、W 0 3 パレットの「入力」に割付けられる。単語 IN-R が処理経路図 T 1 の W 0 3 パレットの「入力」に割付けられる。

## 【 0 0 8 3 】

処理経路図 T 1 の W 0 4 パレットに割り付けられた単語と同じ単語が、W 0 3 パレットの中間に割り付けられる。単語 OT-R が処理経路図 T 1 の W 0 3 パレットの「中間」に割付けられる。

【0084】

プログラマが定義した単語（WK, WK2）は、処理経路図 t 1 の W02 パレット, W03 パレットの「入力」, W03 パレットの「中間」, W04 パレットにそれぞれ割り付けられる。

【0085】

定義体へのアクセスに必要とされる単語（IN-EOF-F）は、パレットとは別のコントロール・ボックス（C/B）に割り付けられる。

【0086】

このような再生準備情報作成処理（ステップ31）によって、従来プログラムを Lyee の方法論にしたがうプログラムに再構築するための準備が完了する。すなわち、Lyee の方法論にしたがうプログラムは、単語を基本としパレットごとに設けられる基底論理（プログラム）である。従来プログラムの中に埋没している単語とパレットとの関係が明らかになり、基底論理を構築するための素材が揃ったことになる。

【0087】

再生準備情報作成処理（ステップ31）が終わると、解析テーブルの作成処理（ステップ32）に移る。この解析テーブルの作成処理は、従来プログラムの各命令文を Lyee のプログラムに対応付け、その対応付けの結果を解析テーブルにまとめて保管するものである。

【0088】

この処理は従来プログラムの命令文の一つ一つについて次の2段階で行われる。

【0089】

まず、従来プログラムの命令文が Lyee のプログラムのどれに対応するかを決定する。上述したように、Lyee のプログラムには、論理要素、作用要素および位相要素があり、作用要素には経路、コマンド、業務および構造の作用要素がある。

【0090】

どの要素（プログラム）に対応するかが決定したら、次に従来プログラムの命令文が決定したプログラムのフローチャートのどの命令（ブロック）に対応する

かを解析する。

【0091】

Lyeeのプログラムの構成要素を簡単に再度まとめておく。

【0092】

媒体アクセスするのは作用要素である。これは定義体識別子を使用しており、OPEN命令、READ命令、WRITE命令およびCLOSE命令が対応する。

【0093】

各パレットに割付けられた単語は位相要素、論理要素または業務作用要素に関係する。位相要素は画面に属する単語のみであり、それはW03パレットの「入力」とW4パレットに割付けられたものである。論理要素には通常単語と等価単語が関係する。通常単語は順序性以外の条件を必要としないロジックに関係するものであり、等価単語は順序性以外の条件を必要とするロジックに関係する。なお、単語ごとの処理の順序性については、Lyeeの方法論によって単語ごとのLyeeのプログラムの内部に吸収されるので、解析は不要である。

【0094】

図14に示す従来プログラムを一命令文ずつ以下に説明する。

【0095】

「01 OPEN INPUT IN-F.」

「02 OPEN OUTPUT OT-F.」

これは「媒体アクセス」であるので作用要素（コマンド）とする。さらにこの命令は作用要素の「実行処理」（図18参照）の部分であるとする。

【0096】

これらの命令に対応するLyeeのプログラムは図19に示すようになる。

【0097】

「03 READ IN-F AT END→MOVE "1" TO IN-EOF-F.」

この命令も「媒体アクセス」であるので作用要素（コマンド）として決定する。更にこの命令は作用要素（コマンド）の「実行処理」の部分であるとする。

【0098】

作用要素（コマンド）の「実行か？」（図18参照）の部分に対応する命令は従来プログラムには無いがLyeeとして必須であるので、新たに必然的必要単語として例えば「IN-F-READ-CTR」（READ件数）を設け、「IF IN-F-READ-CTR = 0」を生成する。更に、命令文番号05のPERFORM UNTIL IN-EOF-F = "1"の解析の段階でこのUNTIL IN-EOF-F = "1"はCOBOLの文法とLyeeプログラムとの関係法則から作用要素（コマンド）の「実行か？」の部分と決定し、「IF IN-F-READ-CTR = 0 AND IN-EOF-F = 0」を生成する。

# 【0099】

作用要素（コマンド）の「成立か？」の部分（図18参照）に対応する命令は従来プログラムには無いがLyeeとして必須であるので、新たに必然的必要単語として例えば「IN-F-READ-STS」（READの状態）を設け、「IF IN-F-READ-STS = 0」を生成する。作用要素（コマンド）の「成立処理」及び「不成立処理」の部分（図18参照）に対応する命令は従来プログラムには無いがLyeeとしてはこのロジックの存在が妥当との解析になるので、どのようにするか予め決定し、そのロジックを生成する。

# 【0100】

「04 MOVE IN-R TO WK」

「IN-R」はW02パレットとW03パレットの入力に割り付けられた単語であり、「WK」はt1に割り付けられた単語として、REMAKE用プログラムには2.4項までの分析で情報が付加されている。そこで、それぞれの単語を「割付単語識別子」で次のように書き直す。

# 【0101】

「MOVE IN-R [T1W03入力] TO WK [t1W02]」

すなわち、この命令はt1W02に割り付けた単語の論理要素として決定される。論理要素のフローチャートは図2に示すとおりである（「成立？」でYESならば「成立処理」を行う、としてもよい）。

# 【0102】

図2に示すフローチャートにおいて、「空か？」とはこの論理要素を実現する単語が所属するパレット上のその単語識別子をアドレスとするデータフィールド

にデーコードが「無い」というロジックになることが、Lyeeの法則で決定している。そしてこの部分のロジックは現行のプログラムには存在しないので、一律的にLyee側で生成する。「自己生成」は「MOVE」文「計算式」かのいずれかであることがLyeeの法則で決定している。従って、現行プログラムの割付単語を端点とする命令が出現したら、その命令をこの部分に当てはめる。上記の例では、MOVE IN-R [T1W03入力] TO WK [t1W02] が当てはめられる。

## 【0103】

「成立？」は、直前の「自己生成」が成立したかどうかであり、この部分に対応する現行プログラムは、この命令の直前にある現行プログラムの条件判断（IF命令やAT END命令など）が該当する。この例の場合には命令文番号03に存在するREAD命令のAT ENDが「成立？」のロジックとして当てはめる。

## 【0104】

ところで、上記事例の単語（WK）については、同じように解析していくと命令文番号10の「10 MOVE WK2 TO WK」と重複することになる。この場合のWKなる単語の割付はt 1である。そこでこの単語を論理要素としたときの「成立か」に該当する現行プログラムを解析する。

## 【0105】

09 IF WK NOT = WK2 OR IN-EOF-F = "1"

WRITE OT-R.

10 MOVE WK2 TO WK.

が存在する。しかし、このIFは命令文番号10だけに係るIFではなく、命令文番号09のWRITE OT-RにもかかるIFである。そしてOT-Rなる単語の割付はT 1 W 0 4である。ここに「境界単語」なる法則が新たに登場する。すなわち、このIFの作用するパレット（座標とも呼ぶ）はOT-Rなる単語の割付がT 1 W 0 4でありWKなる単語の割付がt 1である。このように一つのIFが異なる座標に作用することは原理上矛盾する。この矛盾を避けるために境界単語を追加して割り付ける必要が生じるのである。

## 【0106】

すなわち、命令文番号10の解析の結果、命令文番号04には更に境界単語が追加

され,

MOVE IN-R[T1W03入力] TO WK[t1W02]

MOVE IN-R[T1W03入力] TO WK[T1W03中間] (追加)

となる。

【0 1 0 7】

「05 PERFORM UNTIL IN-EOF = "1"」

は、作用要素 (READコマンド) に使用済みであり、かつ、Lyeeのプログラムには制御のPERFORMは存在しないので、解析対象外となる。

【0 1 0 8】

「06 READ IN-F AT END」

これは、命令文番号03の命令文と同様に解析される。

【0 1 0 9】

「07 MOVE IN-R TO WK2」

「IN-R」はW 0 2パレットとW 0 3パレットの入力に割り付けられた単語であり、「WK2」はt 1に割り付けられた単語及び「境界単語」として、T 1 W 0 3中間とT 1 W 0 4に割り付けられているので、その割付単語識別子で次のように書き直す。

MOVE IN-R [T1W03入力] TO WK2 [t1W02]

MOVE IN-R [T1W03入力] TO WK2 [t1W03中間]

そして、この書き直された命令が論理要素の自己生成に当てはめられ、直前の条件判断命令 (この場合もAT ENDの条件) が論理要素の「成立?」に当てはめられる。

【0 1 1 0】

「08 MOVE WK TO OT-R」

「OT-R」はW 0 3パレット中間とW 0 4パレットに割り付けられた単語であり、「WK」はt 1に割り付けられた単語及び「境界単語」として、T 1 W 0 3中間とT 1 W 0 4に割り付けられているので、その割付単語識別子で次のように書き直す。

MOVE WK [T1W03中間] TO OT-R [T1W03中間]



MOVE WK [t1W04] TO OT-R [T1W03中間]

そして、この書き直された命令が論理要素の自己生成に当てはめられ、直前の条件判断命令（この場合もAT ENDの条件）が論理要素の「成立か」に当てはめられる。

【0 1 1 1】

ここで、新たな矛盾が発見される。すなわち、同じ単語アドレス；OT-R [T1W03中間]に対するプログラムが重複するのである。これに対処するLyeeの法則として「等価単語」が登場する。

【0 1 1 2】

等価単語が現れた場合には、それぞれの「始点単語」データフィールドにデータコードが存在するか否かを論理要素の「成立か」に当てはめ、予め当てはめられている「成立か」の条件判断（この場合は6行目のAT ENDの条件）を「自己生成」の部分に変更する。

【0 1 1 3】

このような解析の結果は、図20に示す解析テーブルに保管される。命令文番号04の解析結果を特に示したのが図21である。

【0 1 1 4】

次にテンプレートの適用（LyeeALL の実行）に進む（ステップ33）。図2、図18に示すようなLyeeのプログラムフローチャートをプログラミング言語で記述したものであり、かつ上記解析テーブルの内容が代入されるようになっているもの（これをテンプレートと呼ぶ）が既に開発されている。これにLyeeALL というツールで解析テーブルの情報を代入する。

【0 1 1 5】

代入により得られたプログラムが「シナリオ関数プログラムT」である。この時点では「パレット関数」，「パレット連鎖関数」，「データディビジョン」は未完成である。

【0 1 1 6】

さらに同期化データ構造の作成を行う（ステップ34）。

【0 1 1 7】

W 0 2 パレットに割り付けた単語（たとえばIN-R）を単純にレコード項目とするデータベースを決定し併せて、その単語をデータ構造とする連絡節を生成する。

【0 1 1 8】

W 0 4 パレットに割り付けた単語（たとえばOT-R）を単純にレコード項目とするデータベースを決定し併せて、その単語をデータ構造とする連絡節を生成する。

【0 1 1 9】

制御ボックスに割り付けた単語（たとえばIN-E0F-F）を制御ボックスとし、併せて、その単語をデータ構造とする連絡節を生成する。このような単純なアルゴリズムでできたデータ構造を「同期化データ構造」と呼ぶ。

【0 1 2 0】

最後に、「パレット関数」，「パレット連鎖関数」，「データディビジョン」が未完成のシナリオ関数Tと「同期化データ構造」からシナリオ関数の完成版（メタTシナリオ関数と呼ぶ）を生成する（ステップ35）。

【図面の簡単な説明】

【図 1】

Lyeeの方法論にしたがうソフトウェア・モデルの構造を示すブロック図である。

【図 2】

W 0 3 論理要素のプログラムの処理の流れを示すフローチャートである。

【図 3】

W 0 2 論理要素のプログラムの処理の流れを示すフローチャートである。

【図 4】

W 0 4 論理要素のプログラムの処理の流れを示すフローチャートである。

【図 5】

W 0 4 論理要素，W 0 2 論理要素，およびW 0 3 論理のプログラムの一例を示す。

【図 6】

オンライン処理 T 0 の処理経路図の一例を示す。

【図 7】

バッチ処理 T 1 の処理経路図の一例を示す。

【図 8】

オンライン処理 T 0 のシナリオ関数 T 0 を示す。

【図 9】

バッチ処理 T 1 のシナリオ関数 T 1 を示す。

【図 1 0】

従来プログラムを Lyee のプログラムに再生する処理の全体的な流れを示す。

【図 1 1】

再生準備情報作成処理の詳細を示す。

【図 1 2】

(a) は従来プログラム中の集団項目とその集団項目を構成する最小単位との関係を示す記述の一例を示す。(b) は分解された最小単位を用いた命令文を、集団項目を用いた命令文とともに示す。

【図 1 3】

コボルによるプログラムの構成要素を示す。

【図 1 4】

従来法によるコボル・プログラムの具体例を示す。

【図 1 5】

データ項目（単語）の抽出の様子を示す。

【図 1 6】

ラシヨナリティの一般的構造を示す。

【図 1 7】

パレットへの単語の割付状態を示す。

【図 1 8】

作用要素（コマンド）のプログラムの処理の流れを示すフローチャートである。

【図 1 9】

具体的に命令文を適用した作用要素プログラムの例を示す。

【図 20】

解析テーブルの概略を示す。

【図 21】

解析テーブルの一例を示す。

【符号の説明】

a 1, b 1, c 1, d 1, e 1 W04パレット

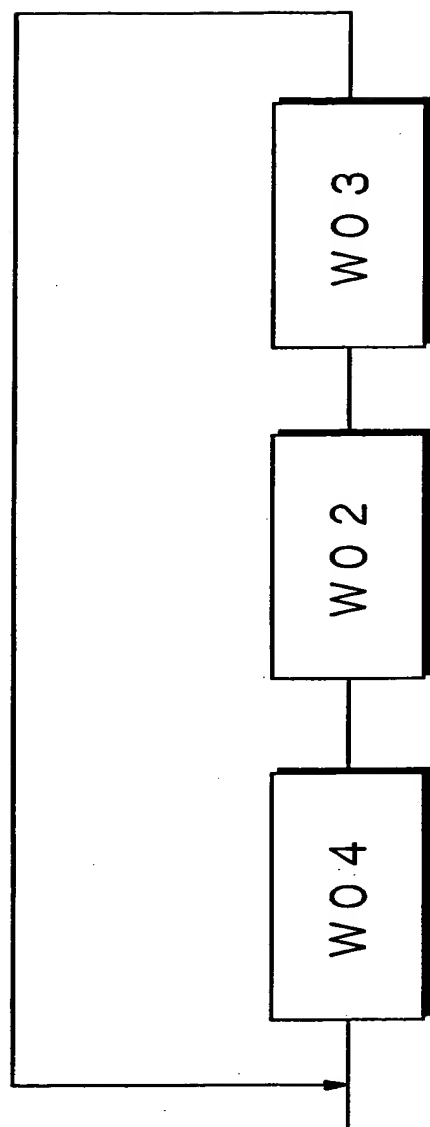
a 2, b 2, c 2, d 2, e 2 W02パレット

a 3, a 4, b 3, c 3, d 3, e 3 W03パレット

【書類名】

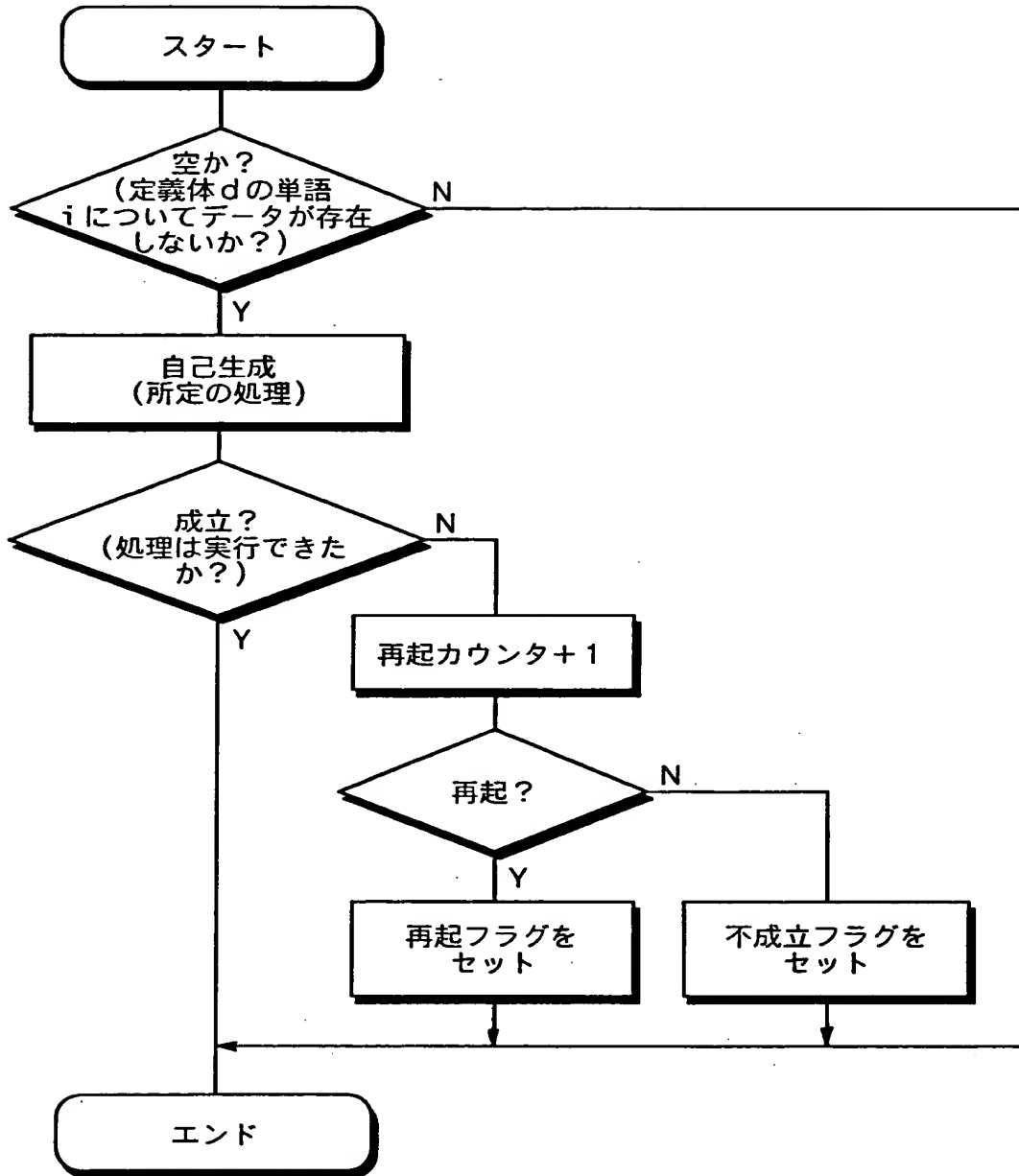
図面

【図 1】

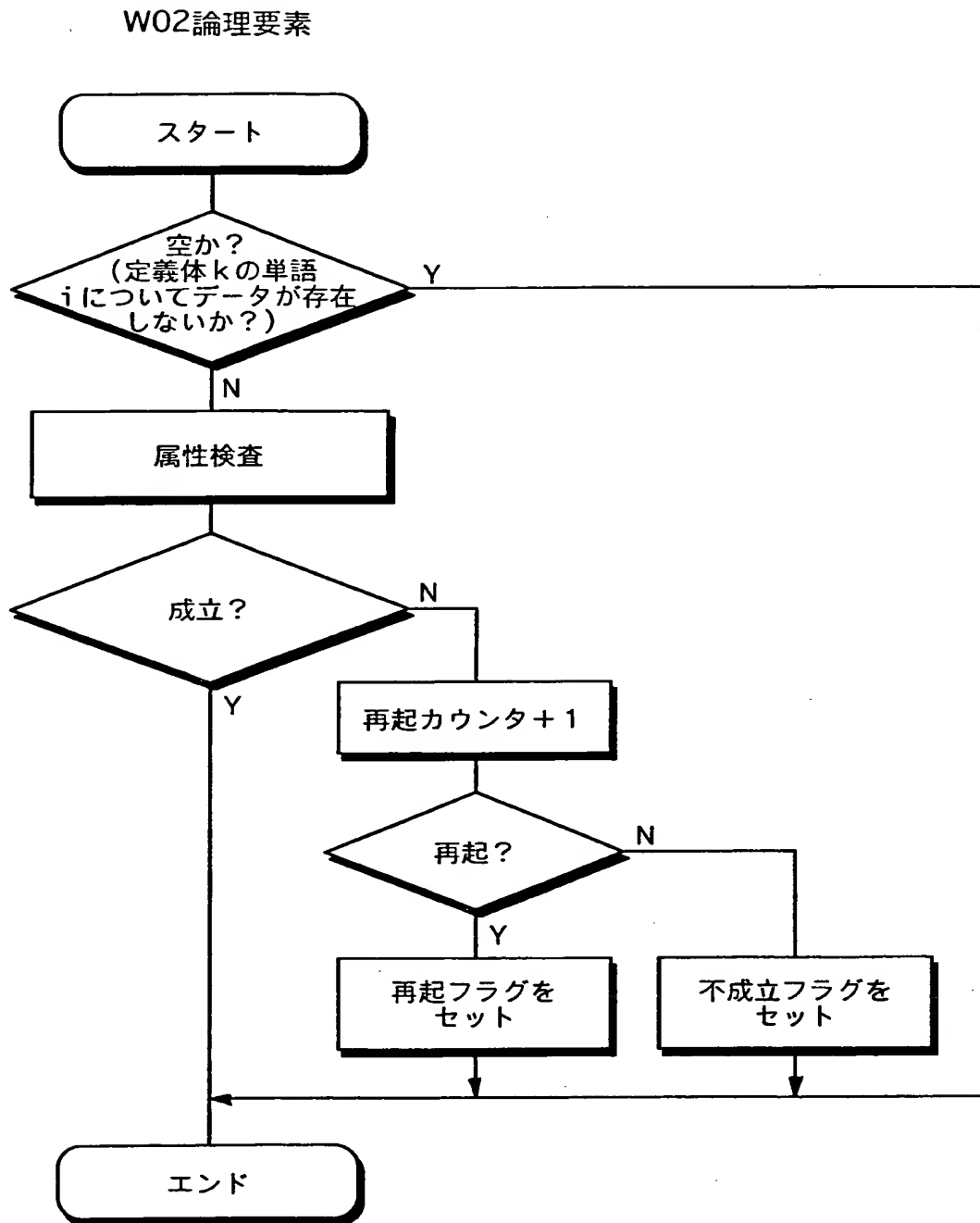


【図 2】

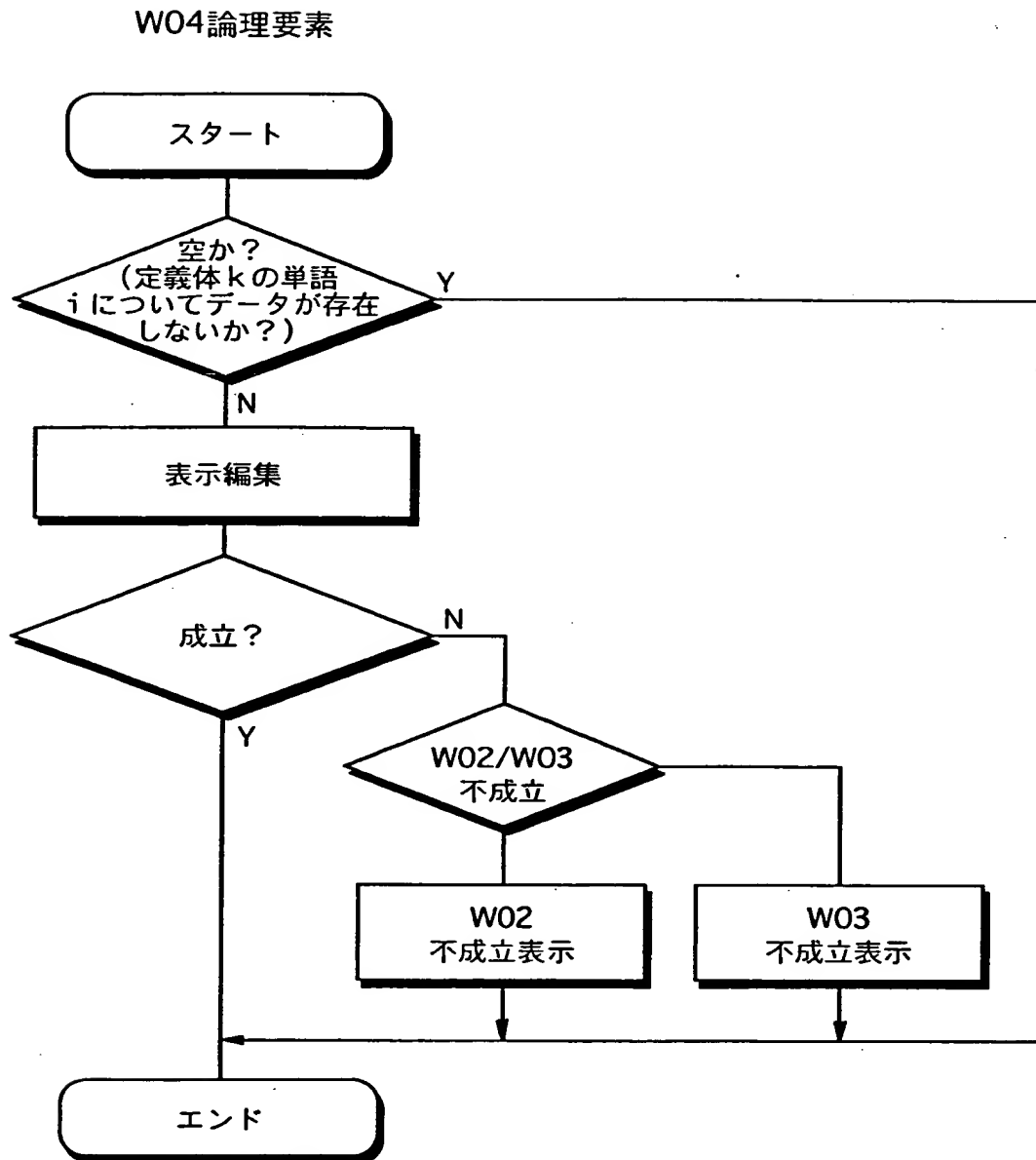
W03論理要素



【図 3】

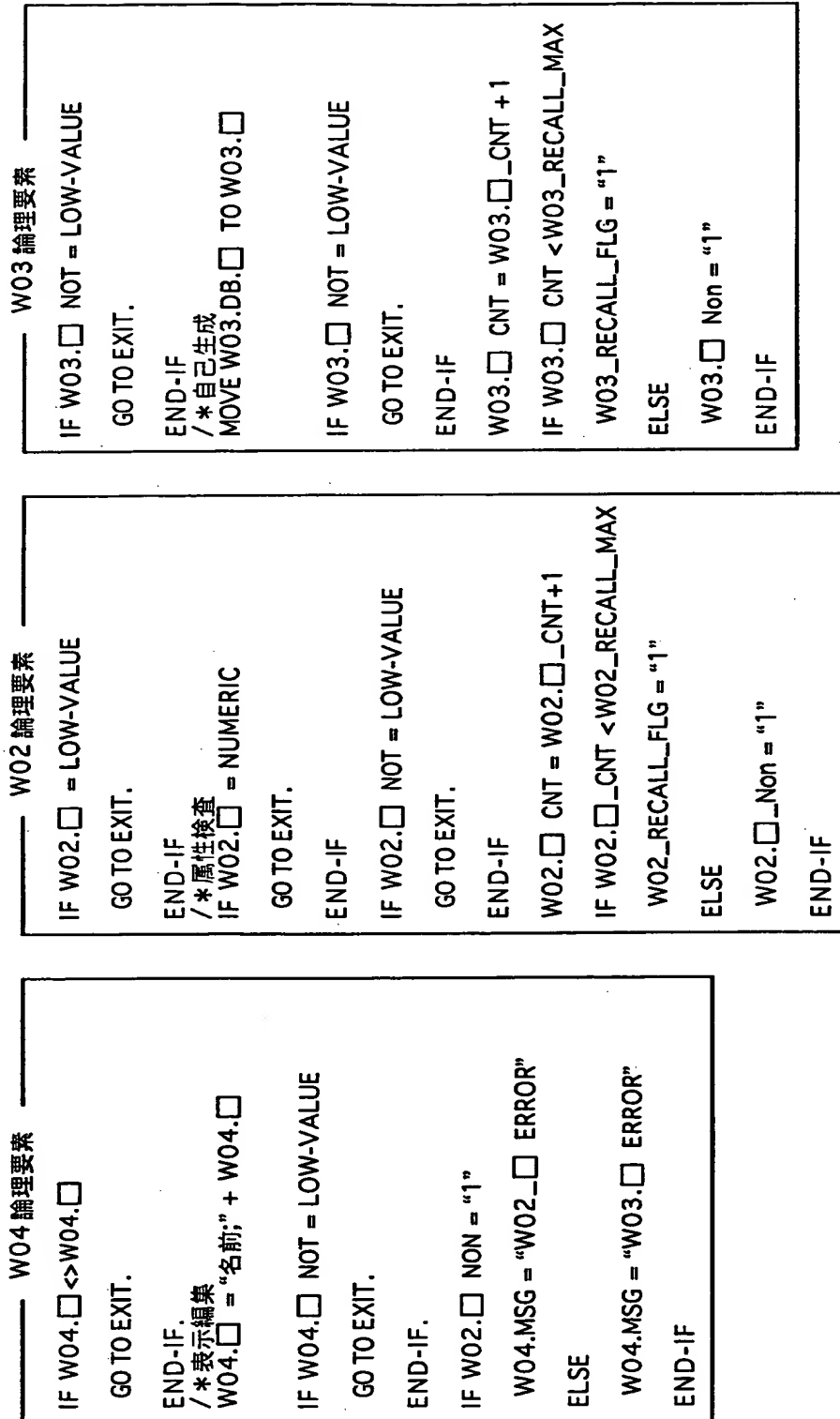


【図 4】

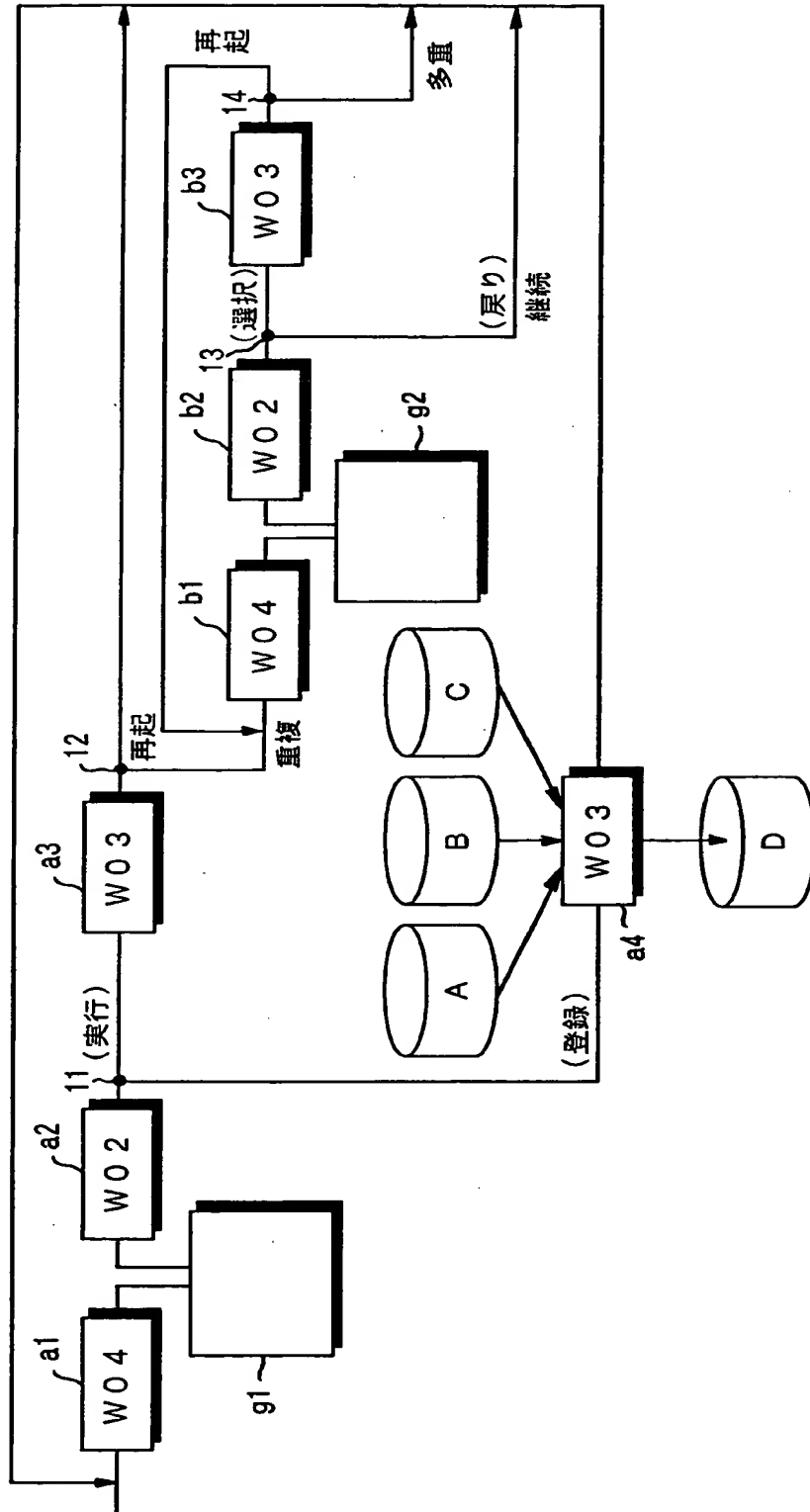




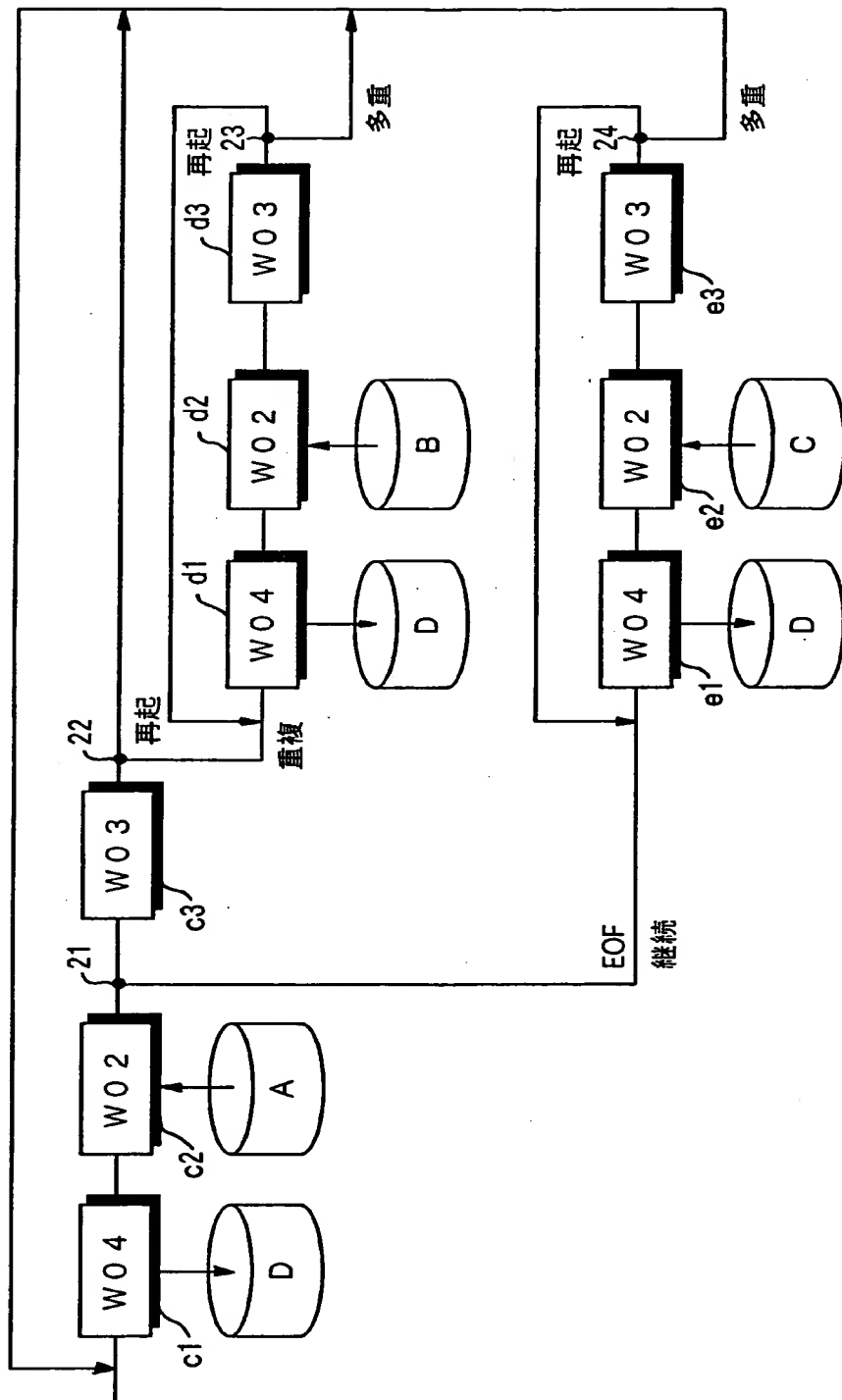
【图 5】



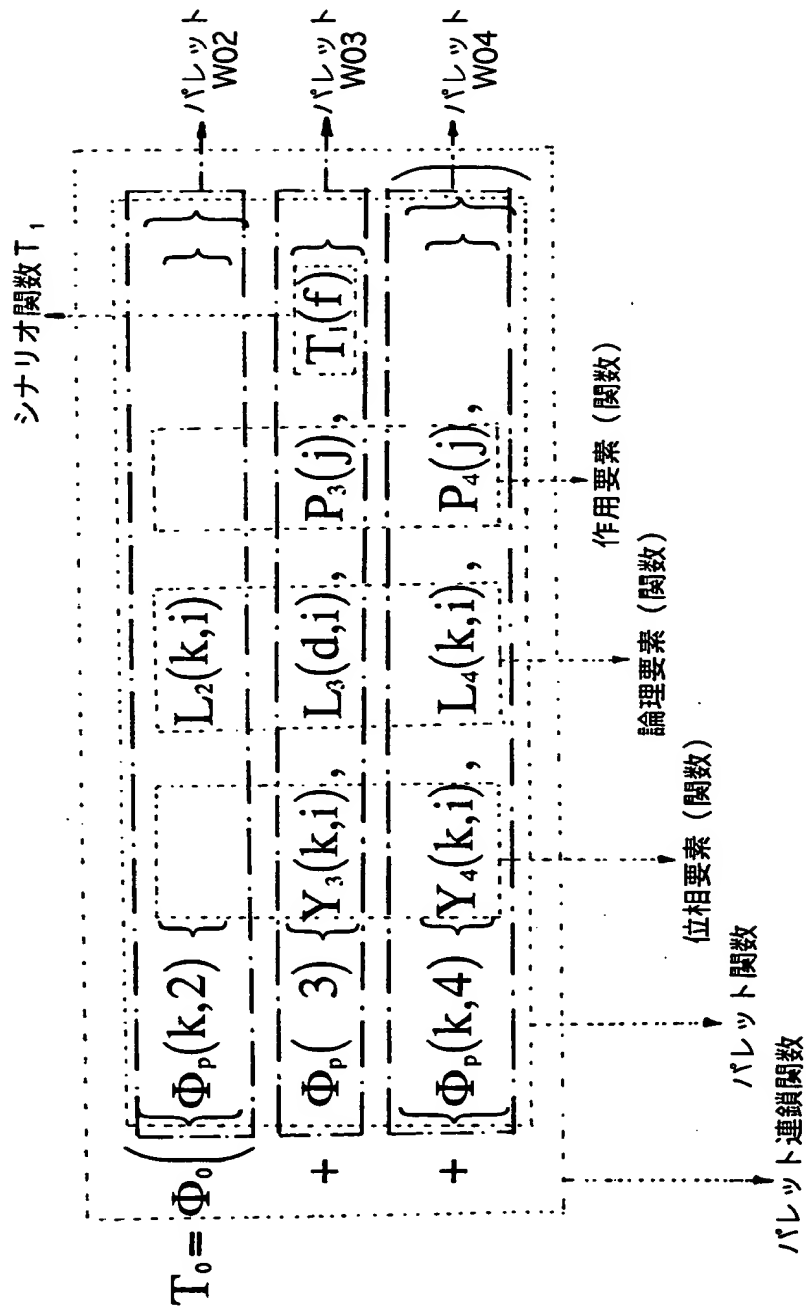
【図 6】



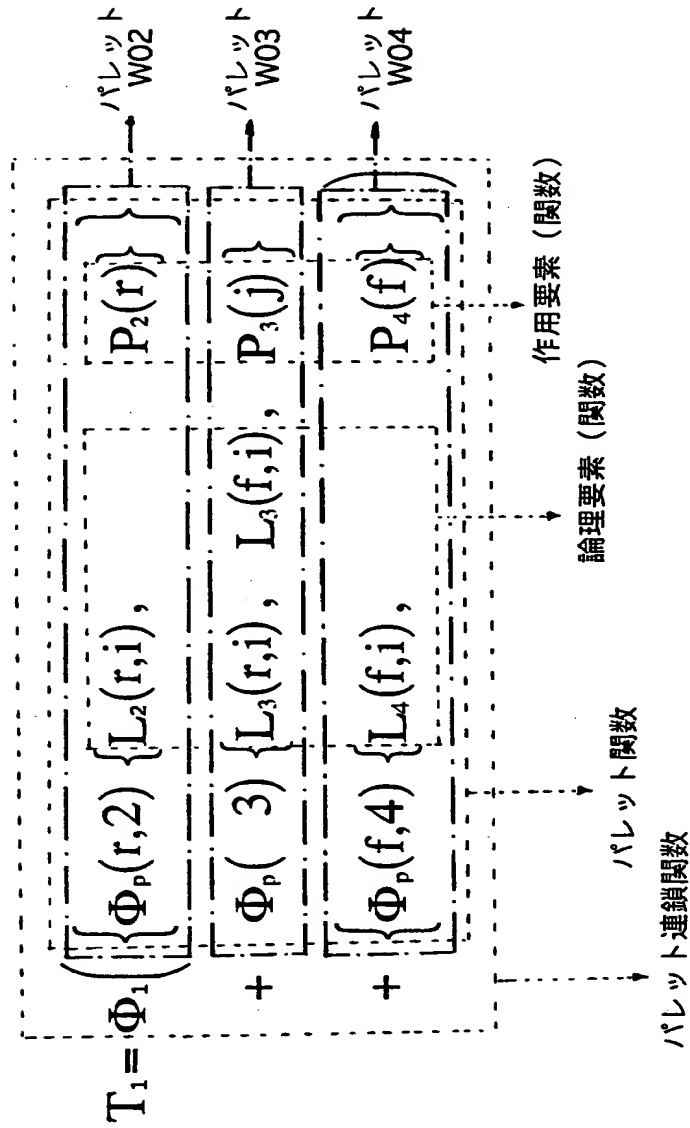
【図 7】



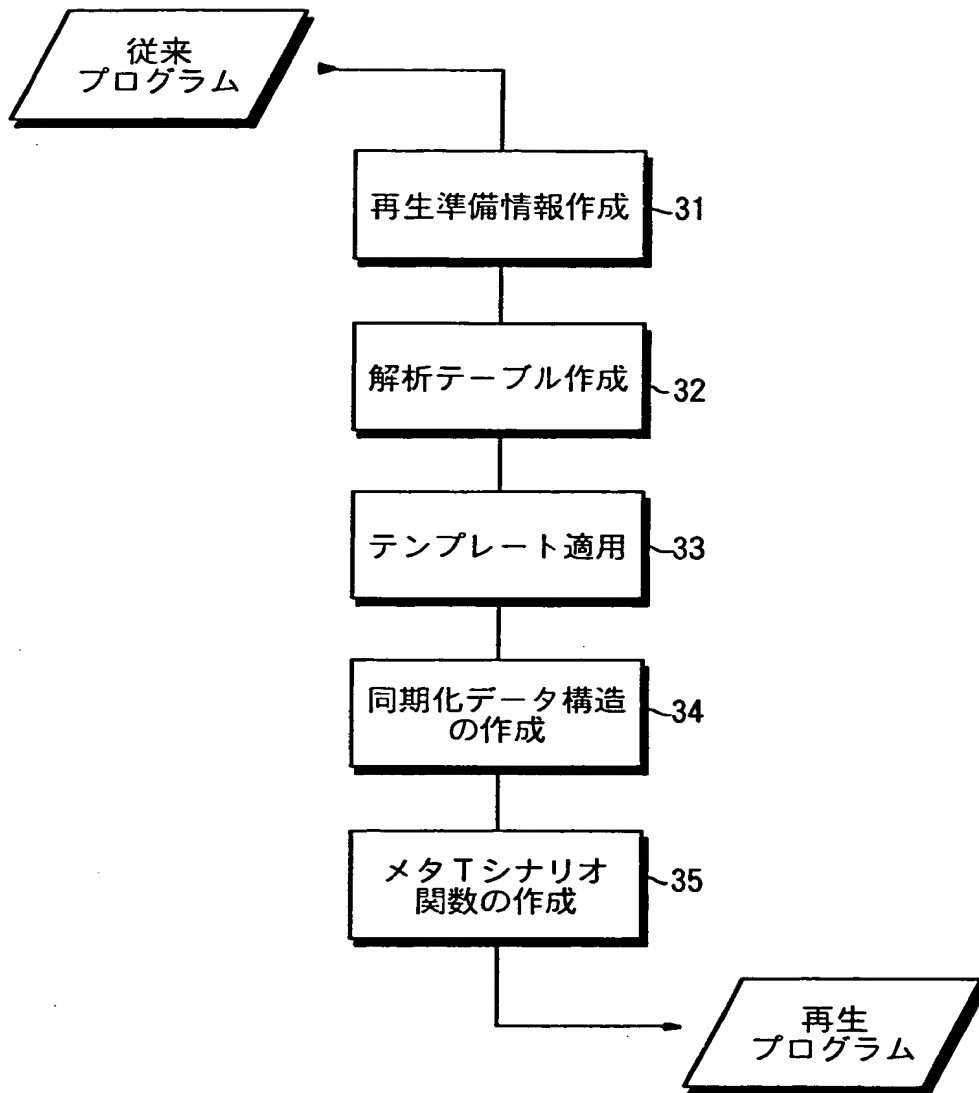
【図 8】



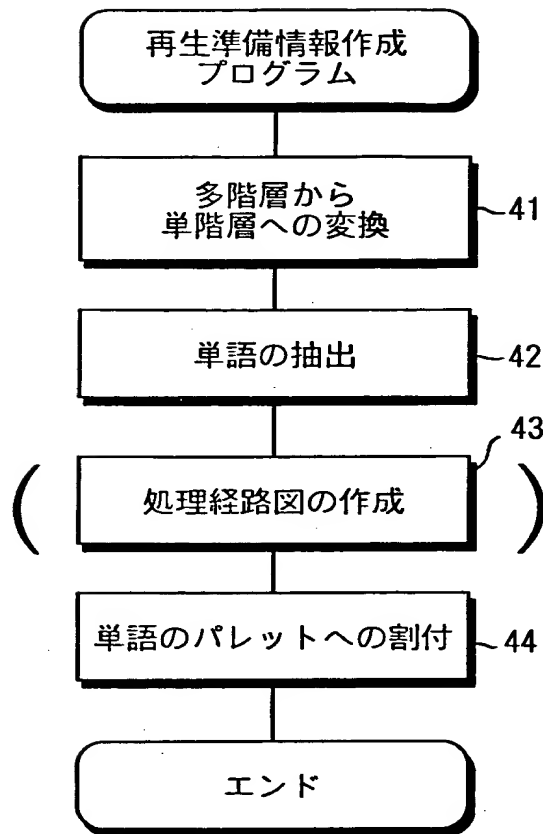
【図 9】



【図 1 0】



【図 1 1】



【図 1 2】

( a )

01 YYMMDD ← 集団項目

03 YY PIC 9(2) ← 最下位レベルのデータ項目

03 MM PIC 9(2) ← 最下位レベルのデータ項目

03 DD PIC 9(2) ← 最下位レベルのデータ項目

( b )

MOVE ZERO TO YYMMDD → { MOVE ZERO TO YY  
MOVE ZERO TO MM  
MOVE ZERO TO DD

【図 1 3】

プログラム識別子の定義	(PROGRAM-ID DIVISION)
プログラム稼働環境定義	(ENVIROMENT DIVISION)
データフィールドの定義	(DATA DIVISION)
媒体の定義	FILE SECTION
作業領域の定義	WORKING-STORAGE SECTION
連絡節の定義	LIKAGE SECTION
処理手続きの定義	(PROCEDURE DIVISION)



【図 1 4】

```

DATA DIVISION.
FILE SECTION.
FD IN-F.
01 IN-R      PIC X(1).
FD OT-F.
01 OT-R      PIC X(1).

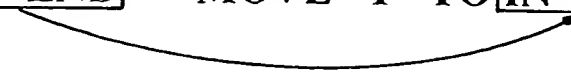
WORKING-STORAGE SECTION.
01 WK        PIC X(1).
01 WK2       PIC X(1).
01 IN-EOF-F  PIC X(1).

PROCEDURE DIVISION.
01 OPEN INPUT IN-F.
02 OPEN OUTPUT OT-F.
03 READ IN-F AT END → MOVE '1' TO IN-EOF-F.
04 MOVE IN-R TO WK.
05 PERFORM UNTIL IN-EOF-F = '1'
06   READ IN-F AT END → MOVE '1' TO IN-EOF-F
07   MOVE IN-R TO WK2
08   MOVE WK TO OT-R
09   IF WK NOT = WK2 OR IN-EOF-F = '1'
      WRITE OT-R
      END-IF
10   MOVE WK2 TO WK
      END-PERFORM.
11 CLOSE IN-F OT-F.

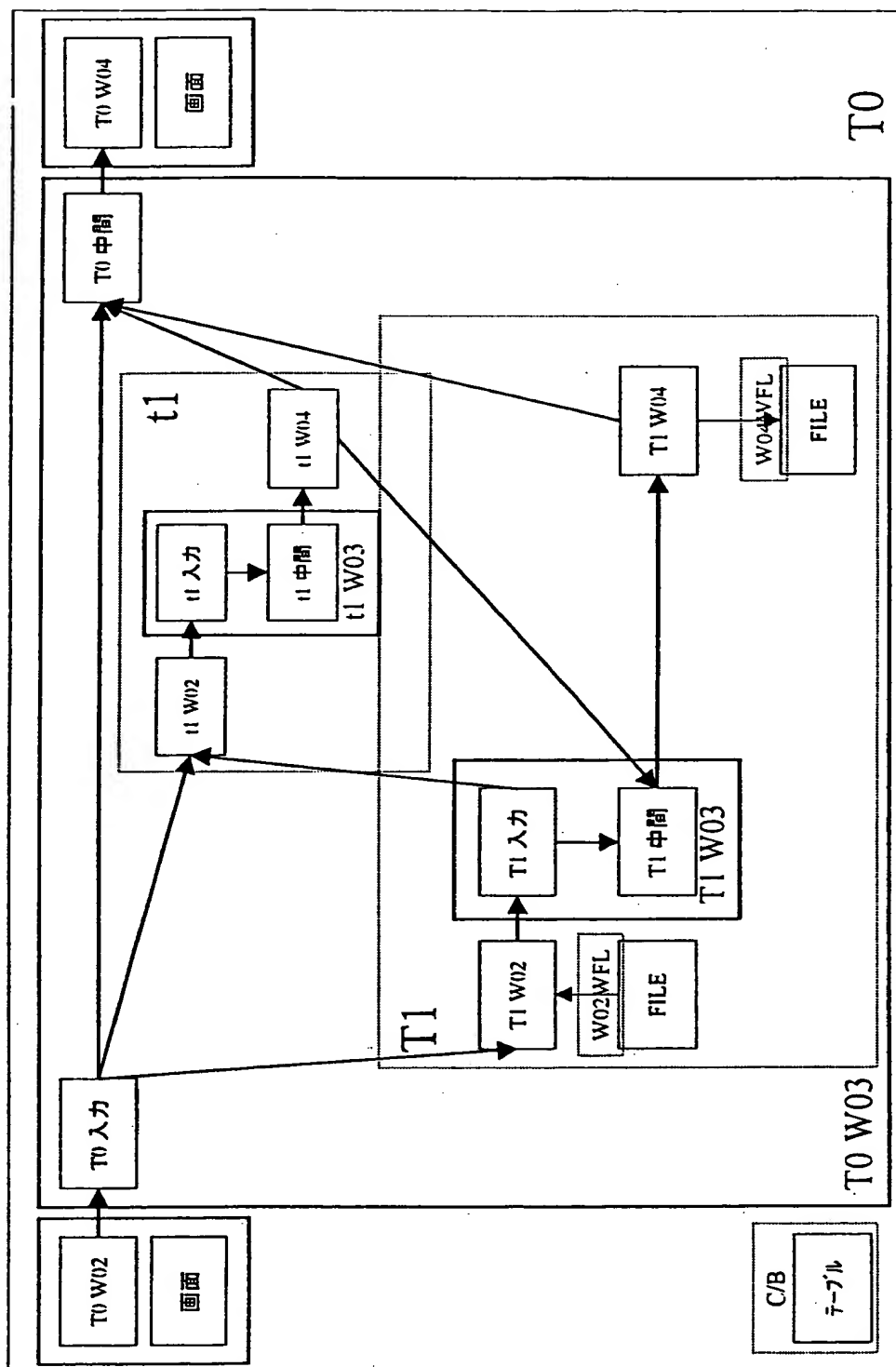
```

【図 1 5】

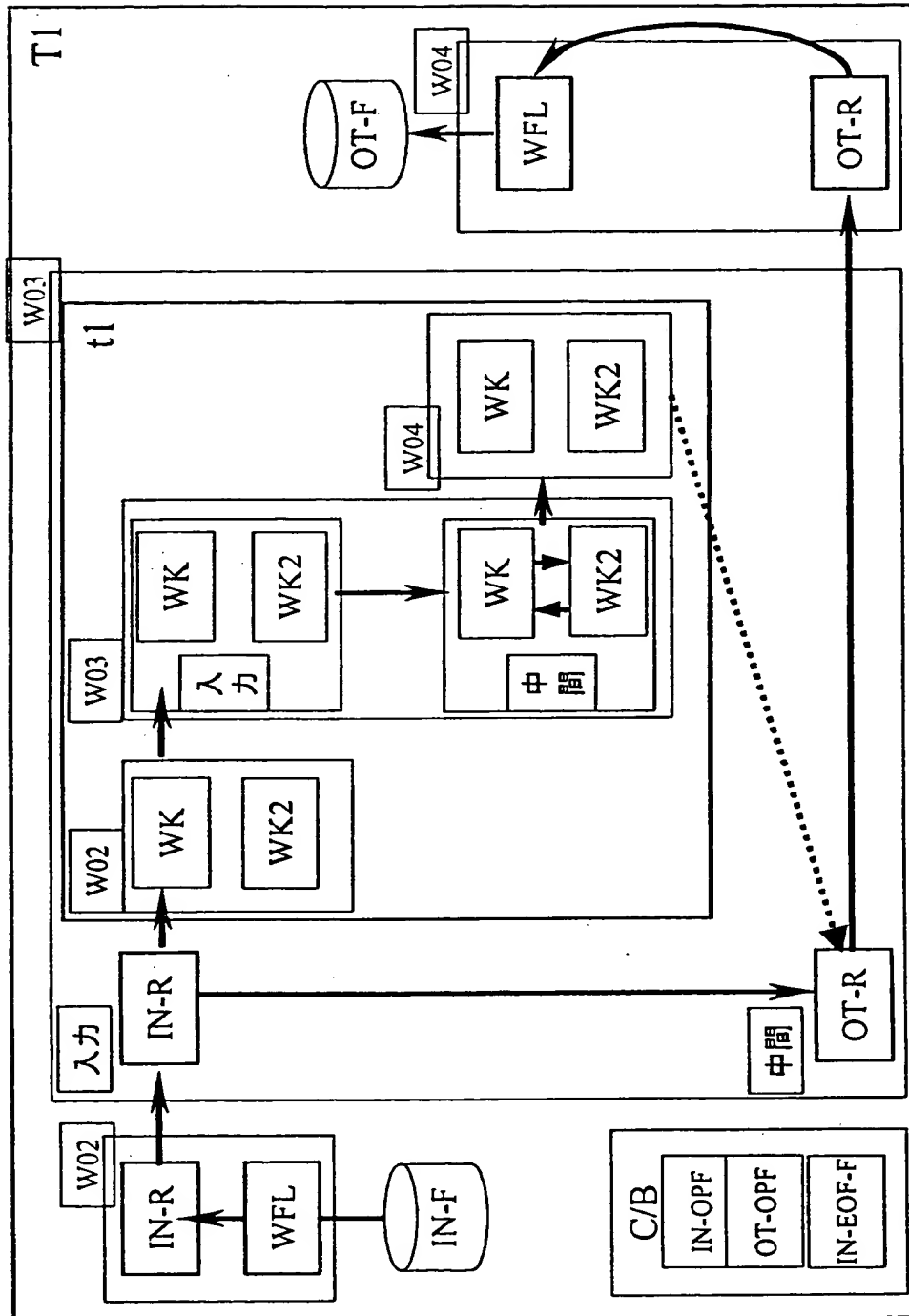
03 READ IN-F **AT END** → MOVE '1' TO **IN-EOF-F**



【図 1 6】

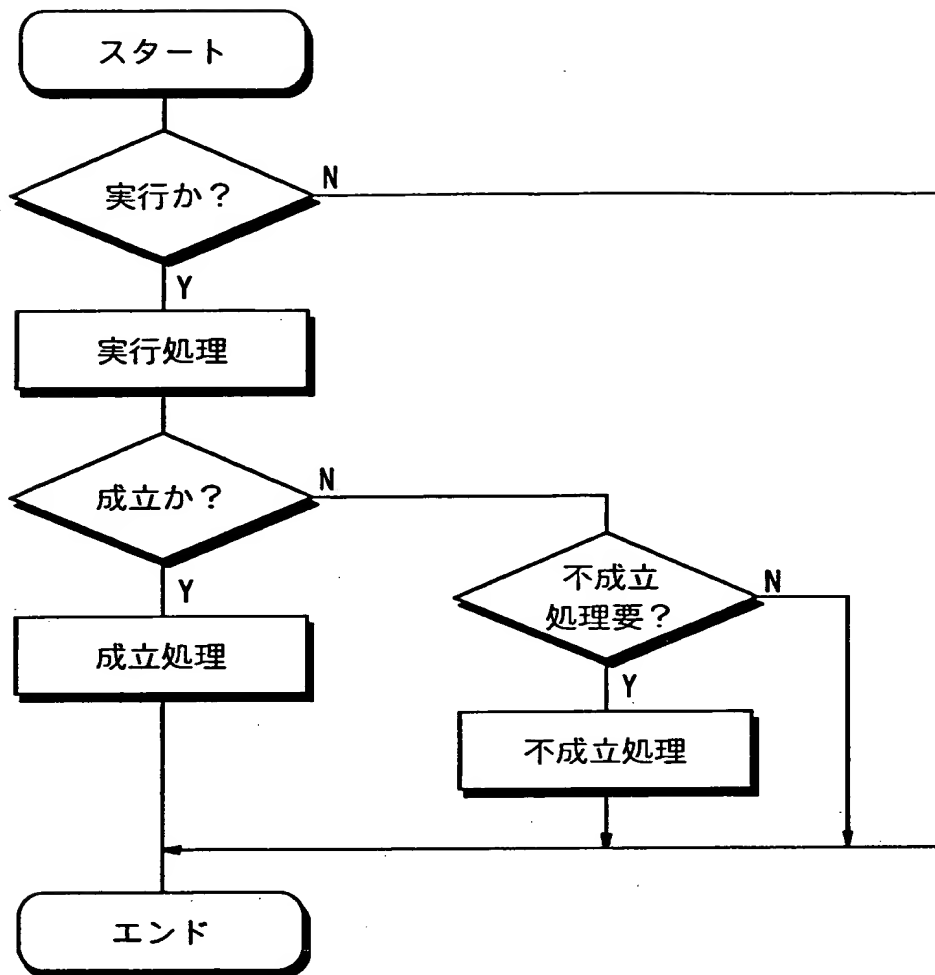


【図 1 7】

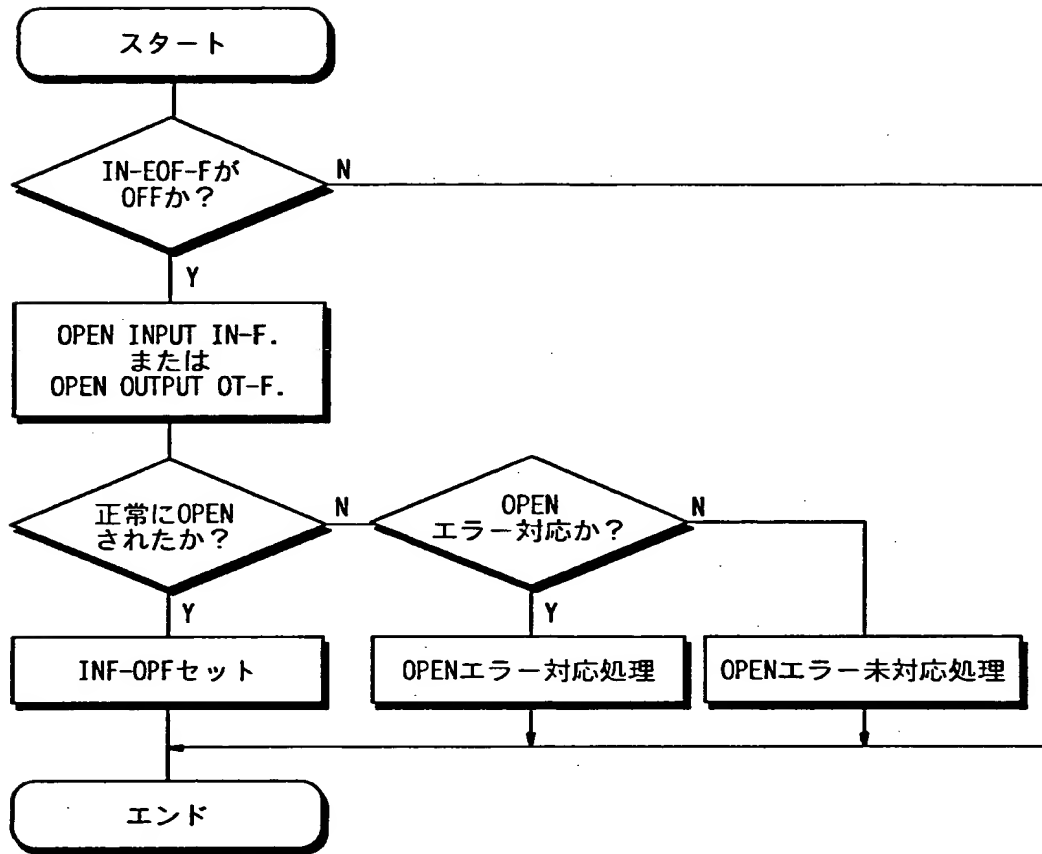


【図 1 8】

作用要素（コマンド）



【図 19】



【図 2 0】

解析結果テーブル

	論理要素	作用要素 (コマンド)	作用要素 (業務)
空の条件 (実行か?)			
自己生成 (実行処理)			
成立条件			
成立処理			
不成立処理			

【図 2 1】

解析テーブル (解析結果情報)

	論理要素 単語識別子 : WK [t1W02]
空の判定条件	IF WK [t1W02] = LOW=VALUE
自己生成	MOVE IN-R [T1W03入力] TO WK [t1W02]
成立の判定条件	IF WK [t1W02] NOT = LOW=VALUE
成立処理	
不成立処理	



【書類名】 要約書

【要約】

【目的】 Lyeeの開発手法以外の手法で作成された既存のプログラムを、その既存のプログラムが達成する機能を維持して、Lyeeの開発手法にしたがう構造をもつ新たなプログラムに比較的容易にかつほぼ自動的に再生する。

【構成】 構成既存のプログラムの記述に基づいて、すべての単語を、それらの単語が関連する定義体を識別した形で抽出し、抽出した単語を、その単語が関連する定義体に応じて、一または複数のパレット（W 0 2，W 0 3，W 0 4）に割付け、既存のプログラムの各命令を、その命令に関連する単語が割付けられたパレットの複数のプログラム構成要素のいずれかにしたがう形に変換する。

【選択図】 図10

出 願 人 履 歴 情 報

識別番号 [598153401]

1. 変更年月日	1998年10月 2日
[変更理由]	新規登録
住 所	東京都港区高輪三丁目11番3号
氏 名	株式会社アイエスデー研究所

出 願 人 履 歴 情 報

識別番号

[599086238]

1. 変更年月日 1999年 6月21日

[変更理由] 新規登録

住 所 東京都港区高輪3丁目11番3号

氏 名 ソフトウェア生産技術研究所株式会社

**THIS PAGE BLANK (USPTO)**